REMARKS

Claims 1-20 are pending in the application. Claims 1-20 have been rejected. Applicants have amended Claim 9 and cancelled Claims 1-8 and 10-20.

Claims 1-4, 9-12, and 15-18 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Fleeson, U.S. Patent No. 6,353,846 B1 (Fleeson). Claims 5-7, 13, and 19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Fleeson in view of Warshavsky et al., U.S. Patent No. 6,732,095 B1 (Warshavsky). Claims 8, 14 and 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Fleeson in view of Bradley et al, U.S. Patent No. 6,584,507 B1 (Bradley). These rejections are respectfully traversed.

The present invention, as set forth by independent claim 9, relates to a team-sharing environment, a method for persisting resource properties in an integrated development environment during transitions of data between a user and a team repository. The method includes storing, in a property file in the integrated development environment, , a list of property keys to be persisted, where the property file is accessible by a user for persisting resource properties during transitions of data and their associated resource property values, storing the property keys and values in different property files for different resources, storing in the property file a cache of prior resource property values, searching the property file for returning a list of the property keys and their associated resource property values, qualifying a property key name by appending the property key name to a contributing resource's name, providing an extension point as an application program interface to third party plug-ins for creating a property file for the third party plug-in.

Fleeson generally relates to resource managed electronic systems. Fleeson discloses a property based decision support system for allocating existing resources to implement a functional unit. The system includes a plurality of resource modules, each providing a component function for implementing a portion of the functional unit and having a set of properties associated therewith. A resource module property object defines a set of properties for each of the plurality of resource modules and an evaluation expression for each of the properties. A link object defines a set of required modules having required properties associated

therewith which are necessary to implement the functional unit. A resource management processor is responsive to a user request for accessing the resource module property object and the link object, and processing the evaluation expression to compare the required properties of each of the required modules to the properties of the plurality of resource modules to determine if the properties of the plurality of resource modules are sufficient to implement functional unit. The processor allocates the resource modules to the functional unit in accordance with the set of required modules defined by the link object if the evaluation expression for each of the required properties is satisfied.

Warshavsky discloses a method to convert data between a relational format and an XML document. The method of Warshavsky creates a set of XML Mapping Definition from metadata; selects relational data from a relational application database, and converts the relational data to the XML document using the set of XML Mapping Definition.

When discussing the element of a property file for storing property keys and their associated resource property values, the Examiner sets forth:

> With respect to independent claim 1, Warshavsky clearly teaches in a team sharing environment, an integrated development environment for persisting resource properties during transitions of data between a user and a team repository (see column 4, lines 38-56), the integrated development environment comprising: a property file for storing property keys and their associated resource property values (see column 5, lines 4-30 and Tables 1-3; Note the property keys are the items listed under property name in Table 1-3 and the property values is the data that resides in those fields.). (Office action dated January 11, 2007, page 5.)

The portion of Warshavsky to which the Examiner refers sets forth:

> XML Mapping Definition 114 consists of three entities: Object, Component, and Field. An Object identifies a specific group of tables and a single XML document to be mapped. The Object contains global information, such as the document's root XML element name. Each Object has a set of components where these components are organized in a hierarchy that can have only one root component.

> A Component defines a mapping between a relational table and XML elements. Two XML elements may be specified for the table: one for the individual records and an optional element to group records belonging to the table. A Component contains zero or more fields. A Field defines the mapping between a column in the Component's table to either an XML element or an XML attribute. The fields within a component may map to a hierarchy of elements and attributes in the XML document.

-4-

The XML Mapping Definition 114 may be automatically populated through use of the Metadata Wizard 115. In one embodiment, the Metadata Wizard 115 is an XML Metadata Wizard and the XML portion of the mapping is fixed by the external metadata, but the default relational portion can be defined by the XML Metadata Wizard. The XML Metadata Wizard may either define a simple mapping where each element of the XML document 104 is associated with a table or it may collapse portions of the XML hierarchy to minimize the number of tables needed to hold the data (Warshavsky Col. 5, lines 4 – 30).

However, nowhere in this portion of Warshavsky, nor anywhere else in Warshavsky is there any disclosure or suggestion of an integrated development environment, much less an integrated development environment which comprises a property file for storing property keys and their associated resource property values as disclosed and claimed. These deficiencies of Warshavsky are not cured by Fleeson.

Bradley discloses linking external information to a network management system. A network management system is installed for and executes in association with a managed network. An external application program is identified by defining and storing in a connection file information that describes: the name and location of the program; a position in a menu control tree into which folders and items, which identify functions and options of the external application program, should be displayed and accessed; security roles associated with each folder and item; and other meta-information about the application program and its maker. The information may be stored in a markup format in a connection file. The network management system reads the connection file and integrates the information into its registry and other locations that determine how the network management system operates.

The examiner cites to a number of portions of Bradley including the following portion

A connection between the network management system and a 3rd-party application is established by creating a connection file that stores information about the connection. To create a connection file, a user selects the ADMIN option 204e, Management Connection sub-option 208a, and the "Create" option 216a within control tree 203 (Bradley, Col. 10, lines 25 30).

However, none of these portions of Bradley discloses or suggests where the environment further comprises an extension point for providing an application program interface to third party plug-ins for creating a property file for the third party plug-in as set forth in claims 9.

More specifically, Fleeson, Warshavsky and Bradley, take alone or in combination, do not teach or suggest a method for persisting resource properties in *an integrated development environment* during transitions of data between a user and a team repository for a team-sharing environment, much less such a method that includes storing, *in a property file of the integrated development environment*, a list of property keys to be persisted and their associated resource property values, *the property file being accessible by a user for persisting resource properties during transitions of data*, storing the property keys and values in different property files for different resources, storing in the property file a cache of prior resource property values, searching the property file for returning a list of the property keys and their associated resource property values, *qualifying a property key name by appending the property key name to a contributing resource's name*, and, *providing an extension point as an application program interface to third party plug-ins for creating a property file for the third party plug-in*, all as required by claim 9. Accordingly, claim 9 is allowable over Fleeson, Warshavsky, and Bradley.

CONCLUSION

In view of the amendments and remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the examiner is requested to telephone the undersigned.

The Commissioner is authorized to deduct any additional fees that may be necessary and to credit any overpayment to Deposit Account No. 090461.

I hereby certify that this correspondence is being electronically submitted to the COMMISSIONER FOR PATENTS via EFS on April 23, 2008.

/Stephen A. Terrile/

_____
Attorney for Applicant(s)

Respectfully submitted,

/Stephen A. Terrile/

Stephen A. Terrile
Attorney for Applicant(s)
Reg. No. 32,946